**Final Project**

Azure Batch

**Problem Statement:**

The opioid crisis is a nation-wide issue with far reaching impact across all spectrums of society. There are many approaches being taken to try to deal with this epidemic and to identify bad actors whose actions contribute greatly to the problem. One area where fraud and abuse can happen is within pharmacies where opioid drugs may be diverted or over-ordered. Unless the companies that own the pharmacies are watching for such activity, it can be missed. While pharmaceutical distributors can also watch for excessive ordering, many pharma customers use multiple suppliers and so any one Supplier may not catch illicit activity. My project uses Azure Batch’s highly scalable compute environment to capture drug ordering data from all sources for a customer and provide that data to an analytic environment where aggregated data across all Suppliers can be used to identify potential opioid fraud and abuse.

**Overview of the Technology:**

To solve the above problem, I created an Azure Batch application that parses EDI Purchase Order Acknowledgment documents to extract drug ordering data, augment that data with demographic and drug classification data, and then store that data in a repository where it can be reported on and analyzed. The Azure Batch application is a Java application based on the Azure Batch Java SDK example but adds a multi-threaded scalable processing environment to take advantage of Azure Batch’s ability to scale to large processing volumes. This solution also uses an Azure-hosted MySQL database for storage of historical ordering data as well as the data needed to augment the raw ordering information.

**High Level Steps:**

1. Download data files and pre-process to produce MySQL insert scripts
2. Install and configure software (Azure Batch Java SDK, local MySQL database)
3. Run Azure Batch example to become familiar with capabilities
4. Created Azure Batch application to parse EDI files, augment drug data, and store to MySQL database
5. Created Python plot.ly script to produce bubble chart from aggregated ordering data
6. Create Azure Batch environment (account, pools, VMs, storage account)
7. Install Java on Azure VMs, install MySQL on Azure VM
8. Execute the Azure Batch application

**Data Sources:**

USPS Zip Code file for State and Zip code data, including latitude and longitude

<https://www.aggdata.com/node/86>

FDA public drug files

[https://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm](http://cp.mcafee.com/d/1jWVIi6x0Sy-yUqemkSkSjpJ55d5VBdZZ4SyyCyYOCUOyrdCPqJyLQFm7bCSm461l4sgp5hO6-CUZUGg8czxlyWaoHFEVsYlIWy_GaeEcK6zASrdCPpesRG9pxjDkdUzkPwt5Izj0S3JoGXelo1y0t5Izj6XaEI9IcLs0gSrjoudCXCQPrNKVJUSyrh)

<https://www.accessdata.fda.gov/cder/ndctext.zip>

EDI files obtained from the MPTS EnterpriseRx pharmacy management system (these were cleansed to remove any customer or supplier references)

**Hardware Used:**

Development: Windows 7 64 bit processor laptop

Azure: Virtual Machines running Ubuntu 16.4 for the MySQL server and for the Azure Batch Pool

**Software Used:**

Java JDK 1.8 (<https://www.oracle.com>)

Azure Batch Java SDK (<https://github.com/Azure/azure-batch-samples/tree/master/Java>)

MySQL 5.7.21 (https://www.mysql.com/)

Plot.ly Python plotting library (https://plot.ly/python/)

**YouTube Links:**

2 Min: <https://www.youtube.com/watch?v=4VFgUvS8UbU>

15 Min: <https://www.youtube.com/watch?v=ZiUikK0qqPE>